The List is Our Process!
An analysis of the kernel's email-based development process

Ralf Ramsauer*, Sebastian Duda†, Lukas Bulwahn‡, Wolfgang Mauerer*§

* Technical University of Applied Sciences Regensburg
† Friedrich–Alexander University Erlangen–Nürnberg
‡ Hobbyist, active in LF ELISA Project, employed at BMW AG
§ Siemens AG, Corporate Research and Technology, Munich

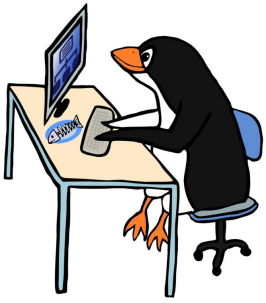
Embedded Linux Conference Europe, Lyon

October 28, 2019

OSTBAYERISCHE
TECHNISCHE HOCHSCHULE
REGENSBURG

Our Overall Goal

Formalising and assessing the Linux Kernel development process



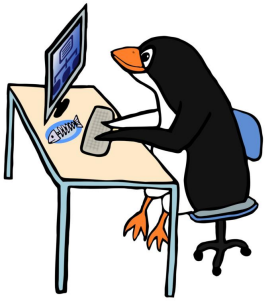
© Moini
openclipart.org

Our Overall Goal

Formalising and assessing the Linux Kernel development process

Outside / Inside Motivation

- Safety-Critical Development
- Development Process Assessment
- Monitoring (cf. CHAOSS)
- Fundamentals of Software Engineering

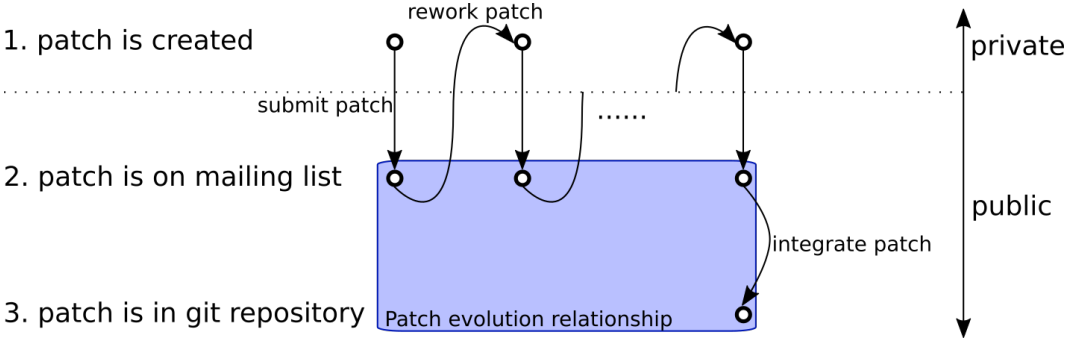


Motivation from *inside* the community

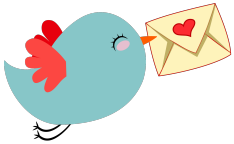
Interest of the kernel community itself

- D. Williams, Towards a Linux Kernel Maintainer Handbook, LPC 2018
- J. Corbet, Change IDs for kernel patches, `https://lwn.net/Articles/797613/`
- [Ksummit-discuss] [MAINTAINERS SUMMIT] Patch version changes in commit logs?
- [Ksummit-discuss] Allowing something Change-Id (or something like it) in kernel commits

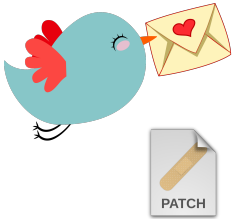
Towards a formal model of the development process



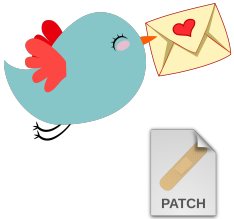
Linux Kernel development workflow



Linux Kernel development workflow

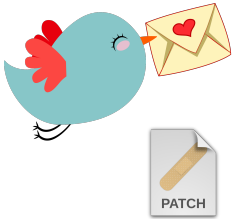


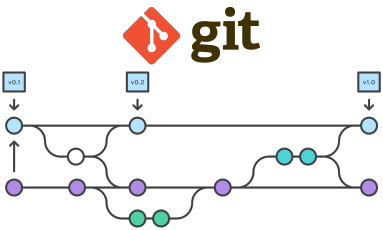
Linux Kernel development workflow



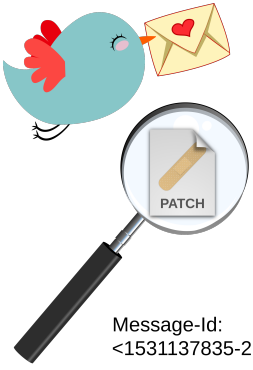
git

Linux Kernel development workflow

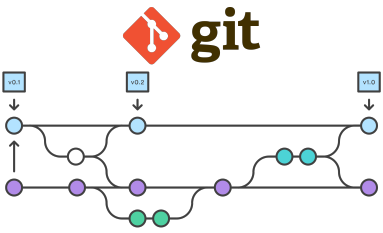




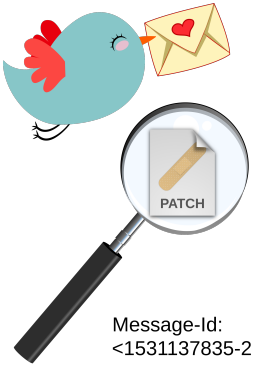
Linux Kernel development workflow



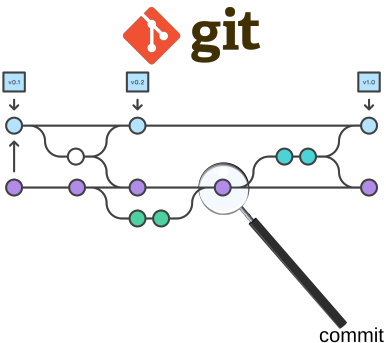
Message-Id:
<1531137835-21581-1-git@1wt.eu>



Linux Kernel development workflow

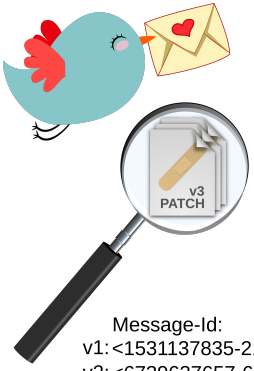


Message-Id:
<1531137835-21581-1-git@1wt.eu>

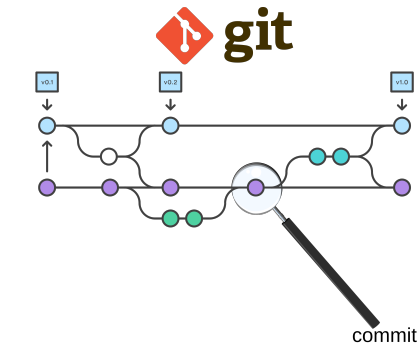


commit
2f6e24d3151fb9967774f9721b288f216f3180df

Linux Kernel development workflow

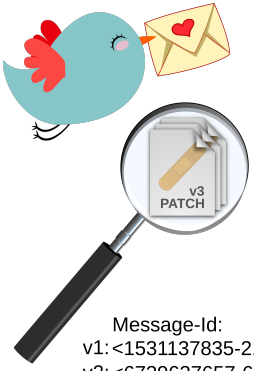


Message-Id:
v1: <1531137835-21581-1-git@1wt.eu>
v2: <6739637657-68462-1-git@1wt.eu>
v3: <9717683099-75474-1-git@1wt.eu>

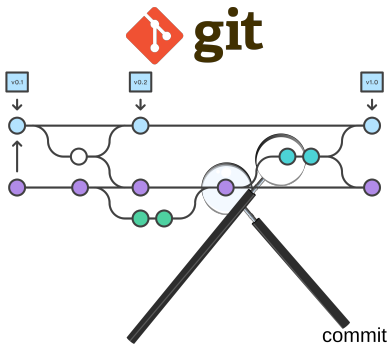


commit
2f6e24d3151fb9967774f9721b288f216f3180df

Linux Kernel development workflow

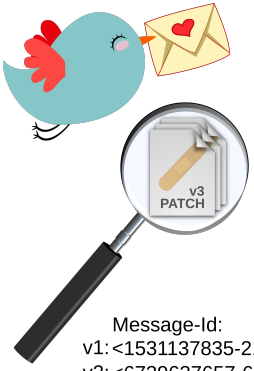


Message-Id:
v1: <1531137835-21581-1-git@1wt.eu>
v2: <6739637657-68462-1-git@1wt.eu>
v3: <9717683099-75474-1-git@1wt.eu>

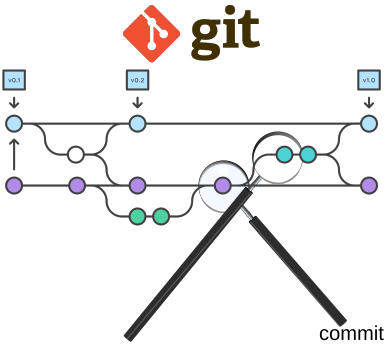


commit
2f6e24d3151fb9967774f9721b288f216f3180df
89876f275e8d562912d9c238cd888b52065cf25

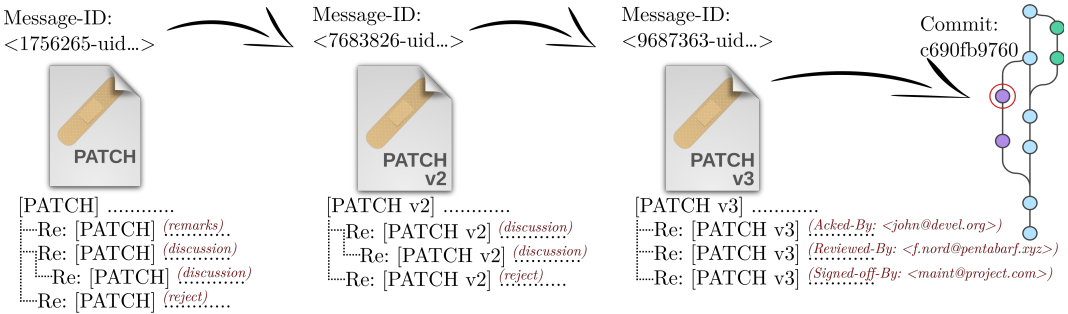
Linux Kernel development workflow



Message-Id:
v1: <1531137835-21581-1-git@1wt.eu>
v2: <6739637657-68462-1-git@1wt.eu>
v3: <9717683099-75474-1-git@1wt.eu>



commit
2f6e24d3151fb9967774f9721b288f216f3180df
89876f275e8d562912d9c238cd888b52065cf25


Message-ID:
<1756265-uid...>
PATCH
[PATCH] .....
Re: [PATCH] (remarks).....
Re: [PATCH] (discussion).....
Re: [PATCH] (discussion).....
Re: [PATCH] (reject).....
Message-ID:
<7683826-uid...>
PATCH v2
[PATCH v2] .....
Re: [PATCH v2] (discussion).....
Re: [PATCH v2] (discussion).....
Re: [PATCH v2] (reject).....
Message-ID:
<9687363-uid...>
PATCH v3
[PATCH v3] .....
Re: [PATCH v3] (Acked-By: <john@devel.org>)
Re: [PATCH v3] (Reviewed-By: <f.nord@pentabarf.xyz>)
Re: [PATCH v3] (Signed-off-By: <maint@project.com>)
Commit:
c690fb9760

PaStA - Patch Stack Analysis

- Detects *similar* patches across different branches
- Quantify mainlining efforts of off-tree developments (Preempt_RT, vendor kernels, . . .)
- Works with mailing lists!

Source: toplock.net.au

Example of similar patches

```
commit 91824d74d6d85f58c63a66b8f2c7993ae246181b
Author: Thomas Gleixner <tglx@linutronix.de>
Date:   Mon Sep 12 21:45:49 2011 +0200

sched-cure-utter-idle-accounting-madness.patrch

Signed-off-by: Thomas Gleixner <tglx@linutronix.de>

diff --git a/kernel/sched.c b/kernel/sched.c
index 205499a..1121a97 100644
--- a/kernel/sched.c
+++ b/kernel/sched.c
@@ -5037,7 +5037,13 @@ EXPORT_SYMBOL(task_nice);
  */
 int idle_cpu(int cpu)
 {
-       return cpu_curr(cpu) == cpu_rq(cpu)->idle;
+       struct rq *rq = cpu_rq(cpu);
+
+#ifdef CONFIG_SMP
+       return rq->curr == rq->idle && !rq->nr_running && !rq->wake_list;
+#else
+       return rq->curr == rq->idle && !rq->nr_running;
+#endif
 }

 /**
```

```
commit 908a3283728d92df36e0c7cd63304fd35e93a8a9
Author: Thomas Gleixner <tglx@linutronix.de>
Date:   Thu Sep 15 15:32:06 2011 +0200

sched: Fix idle_cpu()

On -rt we observed hackbench waking all 400 tasks to a single
cpu. This is because of select_idle_sibling()'s interaction
with the new ipi based wakeup scheme.
[..snip..]
Signed-off-by: Thomas Gleixner <tglx@linutronix.de>
Signed-off-by: Peter Zijlstra <a.p.zijlstra@chello.nl>
Link: http://lkml.kernel.org/n/tip-3b30p18b2[...]
Signed-off-by: Ingo Molnar <mingo@elte.hu>

diff --git a/kernel/sched.c b/kernel/sched.c
index 1874c74..4cdc91c 100644
--- a/kernel/sched.c
+++ b/kernel/sched.c
@@ -5138,7 +5138,20 @@ EXPORT_SYMBOL(task_nice);
  */
 int idle_cpu(int cpu)
 {
-       return cpu_curr(cpu) == cpu_rq(cpu)->idle;
+       struct rq *rq = cpu_rq(cpu);
+
+       if (rq->curr != rq->idle)
+               return 0;
+
+       if (rq->nr_running)
+               return 0;
+
+#ifdef CONFIG_SMP
+       if (!llist_empty(&rq->wake_list))
+               return 0;
+#endif
+
+       return 1;
 }

 /**
```

Example of similar patches

commit 91824d74d6d85f58c63a66b8f2c7993ae246181b
Author: Thomas Gleixner <tglx@linutronix.de>
Date: Mon Sep 12 21:45:49 2011 +0200

sched-cure-utter-idle-accounting-madness.patrch

Signed-off-by: Thomas Gleixner <tglx@linutronix.de>

diff --git a/kernel/sched.c b/kernel/sched.c
index 205499a..1121a97 100644

```
+       struct rq *rq = cpu_rq(cpu);
+
+#ifdef CONFIG_SMP
+       return rq->curr == rq->idle && !rq->nr_r
+#else
+       return rq->curr == rq->idle && !rq->nr_r
+#endif
```

}

/**

commit 908a3283728d92df36e0c7cd63304fd35e93a8a9
Author: Thomas Gleixner <tglx@linutronix.de>
Date: Thu Sep 15 15:32:06 2011 +0200

sched: Fix idle_cpu()

On -rt we observed hackbench waking all 400 tasks to a single cpu. This is because of select_idle_sibling()'s interaction with the new ipi based wakeup scheme.
[..snip..]
Signed-off-by: Thomas Gleixner <tglx@linutronix.de>

```
+       struct rq *rq = cpu_rq(cpu);
+
+       if (rq->curr != rq->idle)
+               return 0;
+
+       if (rq->nr_running)
+               return 0;
+
+#ifdef CONFIG_SMP
+       if (!llist_empty(&rq->wake_list))
+               return 0;
+#endif
+
+       return 1;
```

Example of similar patches

```
commit 91824d74d6d85f58c63a66b8f2c7993ae246181b
Author: Th                                    >
Date:   Mo

sched–cure-                                   atrch

Signed–off-                                   onix.de>

 diff ––git                                   c
 index 2054
––– a/kern
+++ b/kern
@@ –5037,7                                    nice);
   */
  int idle_
  {
–         re                                  –>idle;
+         st
+
+#ifdef CO
+         re                                  >nr_running && !rq–>wake_list;
+#else
+         return rq–>curr == rq–>idle && !rq–>nr_running;
+#endif
  }

  /**
```

```
[...]
#else
#endif
#ifdef
CONFIG_SMP
cpu_rq(cpu)
return
rq
rq–>curr
rq–>idle
rq–>nr_running
rq–>wake_list
struct
```

```
commit 908a3283728d92df36e0c7cd63304fd35e93a8a9
Author: Thomas Gleixner <tglx@linutronix.de>
Date:   Thu Sep 15 15:32:06 2011 +0200

sched: Fix idle_cpu()

On –rt we observed hackbench waking all 400 tasks to a single
cpu. Th                                       ing()'s interaction
with the
[..snip
Signed–                                       utronix.de>
Signed–                                       ra@chello.nl>
Link: h                                       18b2[...]
Signed–                                       u>

 diff ––                                      ed.c
 index 18
––– a/k
+++ b/k
@@ –513                                       k_nice);
   */
  int idl
  {
–                                             pu)–>idle;
+
+
+
+
+
+                 return 0;
+
+#ifdef CONFIG_SMP
+         if (!llist_empty(&rq–>wake_list))
+                 return 0;
+#endif
+
+         return 1;
  }

  /**
```

```
[...]
#endif
#ifdef
if
CONFIG_SMP
cpu_rq(cpu)
llist_empty
return
rq
rq–>curr
rq–>idle
rq–>nr_running
rq–>wake_list
struct
```

Example of similar patches

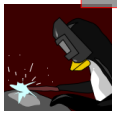
```
commit 91824d74d6d85f58c63a66b8f2c7993ae246181b
Author: Th                                    >
Date:   Mo

sched–cure-                                   atrch

Signed–off-                                   onix.de>

 diff ––git                                   c
 index 2054
––– a/kern
+++ b/kern
@@ –5037,7                                    nice);
   */
  int idle_
  {
–         re                                  –>idle;
+         st
+
+#ifdef CO
+         re                                  >nr_running && !rq–>wake_list;
+#else
+         return rq–>curr == rq–>idle && !rq–>nr_running;
+#endif
  }

  /**
```

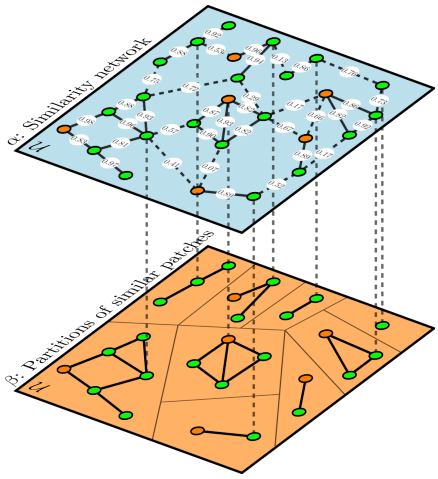
```
[...]
#else
#endif
#ifdef
CONFIG_SMP
cpu_rq(cpu)
return
rq
rq–>curr
rq–>idle
rq–>nr_running
rq–>wake_list
struct
```

```
commit 908a3283728d92df36e0c7cd63304fd35e93a8a9
Author: Thomas Gleixner <tglx@linutronix.de>
Date:   Thu Sep 15 15:32:06 2011 +0200

sched: Fix idle_cpu()

On –rt we observed hackbench waking all 400 tasks to a single
cpu. Th                                       ing()'s interaction
with the
[..snip
Signed–                                       utronix.de>
Signed–                                       ra@chello.nl>
Link: h                                       18b2[...]
Signed–                                       u>

 diff ––                                      ed.c
 index 18
––– a/k
+++ b/k
@@ –513                                       k_nice);
   */
  int idl
  {
–                                             pu)–>idle;
+
+
+
+
+
+                 return 0;
+
+                 return 0;
+#endif
+
+         return 1;
  }

  /**
```

```
[...]
#endif
#ifdef
if
CONFIG_SMP
cpu_rq(cpu)
llist_empty
return
rq
rq–>curr
rq–>idle
rq–>nr_running
rq–>wake_list
struct
```

Diff similarity: 0.875

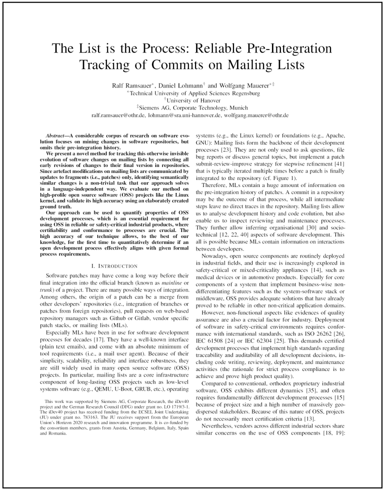




Legend

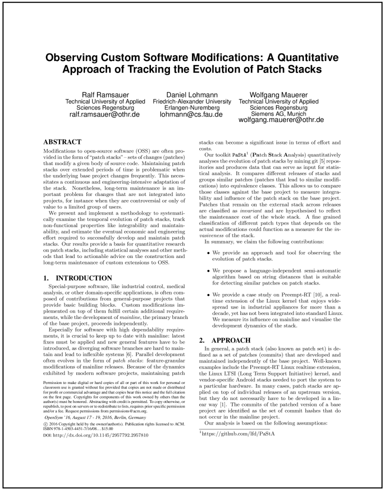
- green nodes: patches on MLs
- orange nodes: commits in repository
- edges: similarity of patches/commits
  - dashed: similarity below thres
  - solid: similarity above thres

Interested in the techniques? More details in:



The List is the Process: Reliable Pre-Integration Tracking of Commits on Mailing Lists

Ralf Ramsauer*, Daniel Lohmann† and Wolfgang Mauerer*‡
*Technical University of Applied Sciences Regensburg
†University of Hanover
‡Siemens AG, Corporate Technology, Munich
ralf.ramsauer@othr.de, lohmann@sra.uni-hannover.de, wolfgang.mauerer@othr.de



Observing Custom Software Modifications: A Quantitative Approach of Tracking the Evolution of Patch Stacks

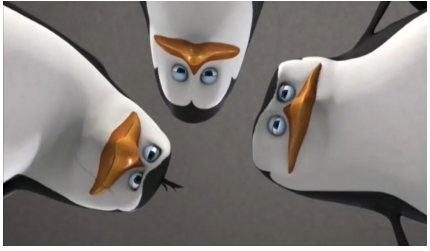
Ralf Ramsauer, Technical University of Applied Sciences Regensburg, ralf.ramsauer@othr.de
Daniel Lohmann, Friedrich-Alexander University Erlangen-Nuremberg, lohmann@cs.fau.de
Wolfgang Mauerer, Technical University of Applied Sciences Regensburg, Siemens AG, Munich, wolfgang.mauerer@othr.de

Data Acquisition

- Dumps from gmane.org etc.
- kernel.org public inboxes
  - some lists, prehistoric data
  - `https://lore.kernel.org/lists.html`
  - Some lists are imports from gmane.org :-(
- Our own collection

Data Acquisition

- Dumps from gmane.org etc.
- kernel.org public inboxes
  - some lists, prehistoric data
  - `https://lore.kernel.org/lists.html`
  - Some lists are imports from gmane.org :-(
- Our own collection
  - 200 lists, since May ’19
  - `https://github.com/linux-mailinglist-archives`



© Penguins of Madagascar
20th Century Fox

Let there be chaos

- Br?ken encoding
- BÃse64
-
- MUAs
- Bots
- HTML
- Automated mails
- non-Linux patches
- Stable reviews
- Malformed recipients
- ...

Message-Id: <74851t0$h3103wn0$Delldi Fri, 9 Mar 71685 18:45:56 +0000

Date: Mon, 08 Aug 05 04:01:15 ?x?_???????

Date: Tue, 27 Mar 22001 13:42:39 +0200 (Westeurop?ische Sommerzeit)

X-Mailer: Microsoft Outlook Express 6.00.2900.3028

We analyse...

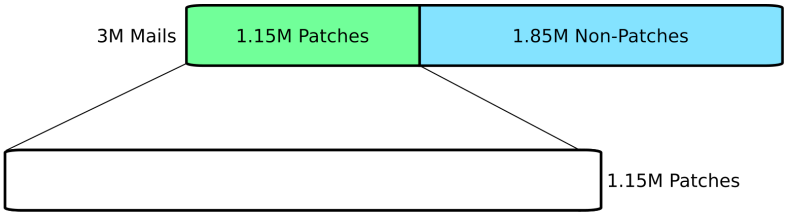
- v2.6.39..linus/master
- ≈610K commits
- Mails: 2011-05-01 – 2018-12-31
- ≈3M mails
- Lists: All Public Inboxes from lore.kernel.org

We analyse...

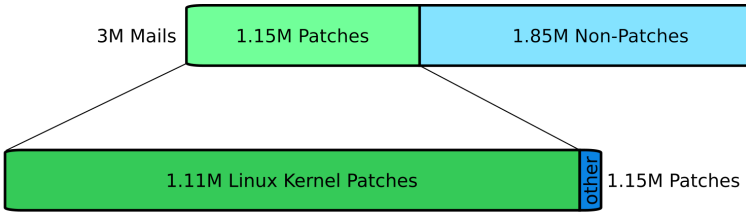
- v2.6.39..linus/master
- ≈610K commits
- Mails: 2011-05-01 – 2018-12-31
- ≈3M mails
- Lists: All Public Inboxes from lore.kernel.org
- linux-amlogic, **linux-arm-kernel**, linux-i3c, linux-mtd, linux-riscv, linuxppc-dev, cocci, linux-block, linux-bluetooth, linux-btrfs, linux-cifs, linux-clk, linux-crypto, linux-ext4, linux-fsdevel, linux-hwmon, linux-iio, linux-integrity, **linux-kernel**, linux-media, linux-mips, linux-modules, linux-next, **netdev**, linux-nfs, linux-parisc, linux-pci, linux-renesas-soc, linux-rtc, linux-security-module, linux-sgx, linux-trace-devel, linux-watchdog, **linux-wireless**

2011-05–2018-12

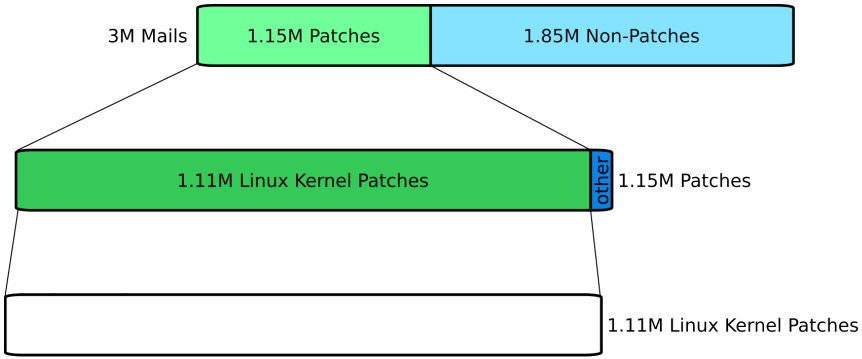
2011-05–2018-12



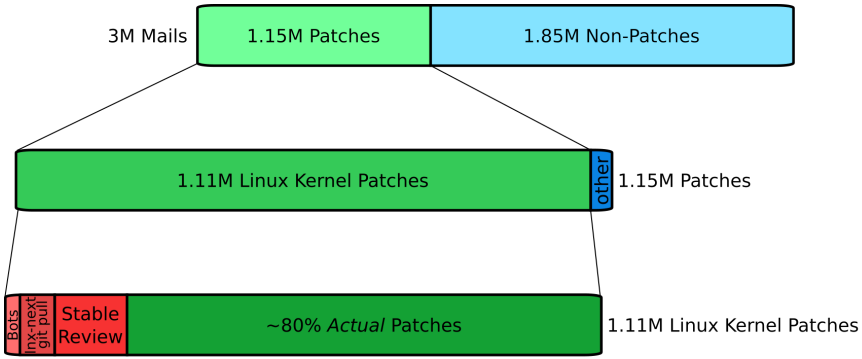
2011-05–2018-12



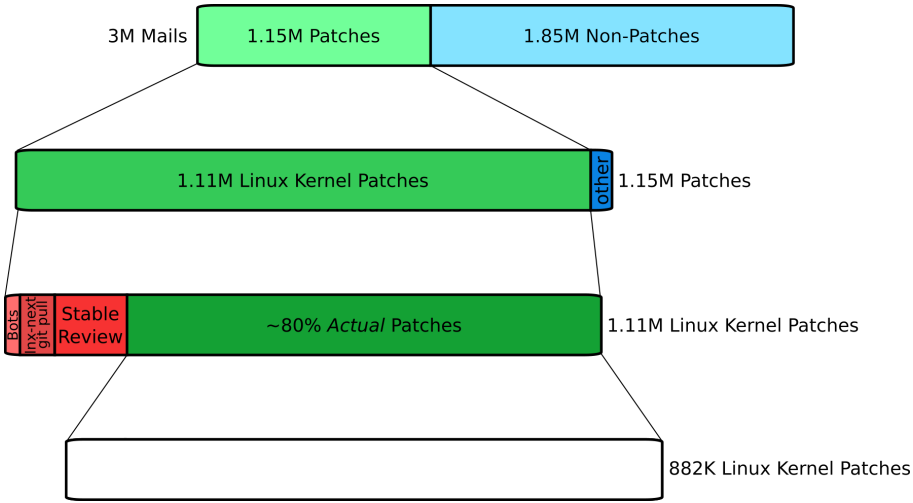
2011-05–2018-12



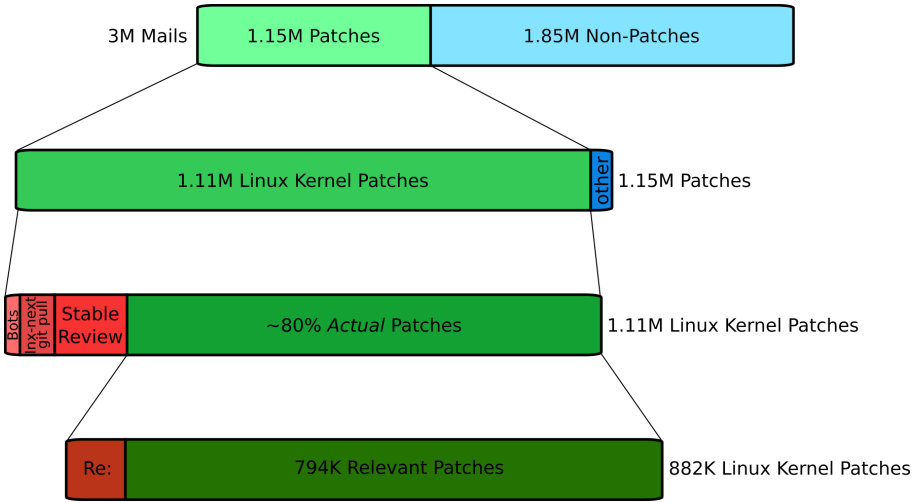
2011-05–2018-12



2011-05–2018-12



2011-05–2018-12



Ignored Patches

Research Question

Are there specific characteristics for ignored patches?

Definition

A patch on a ML is *ignored* if...

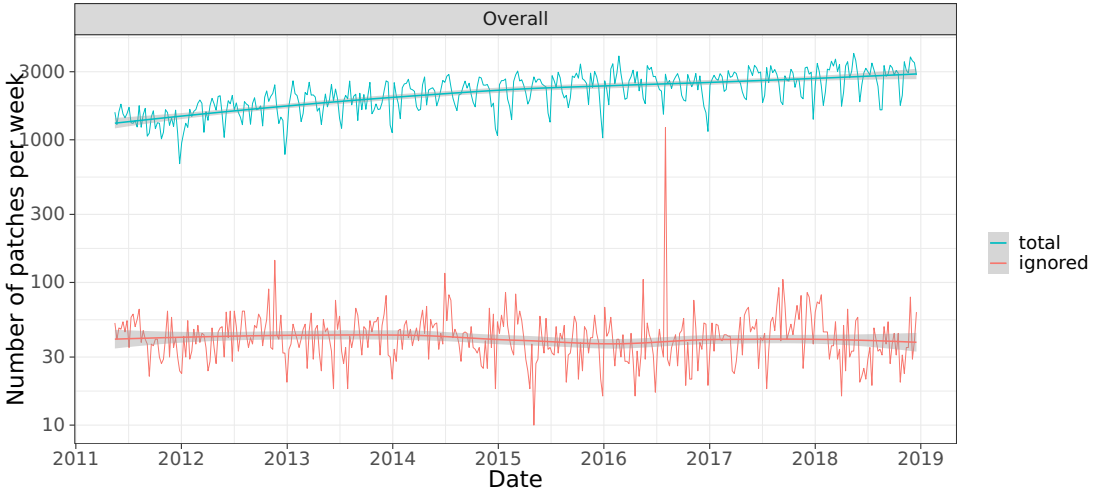
- ... the thread of the patch has no responses from persons other than the author
- ... the patch was not accepted upstream
- ... all related patches (e.g., revisions in other series) were ignored

Ignored Patches

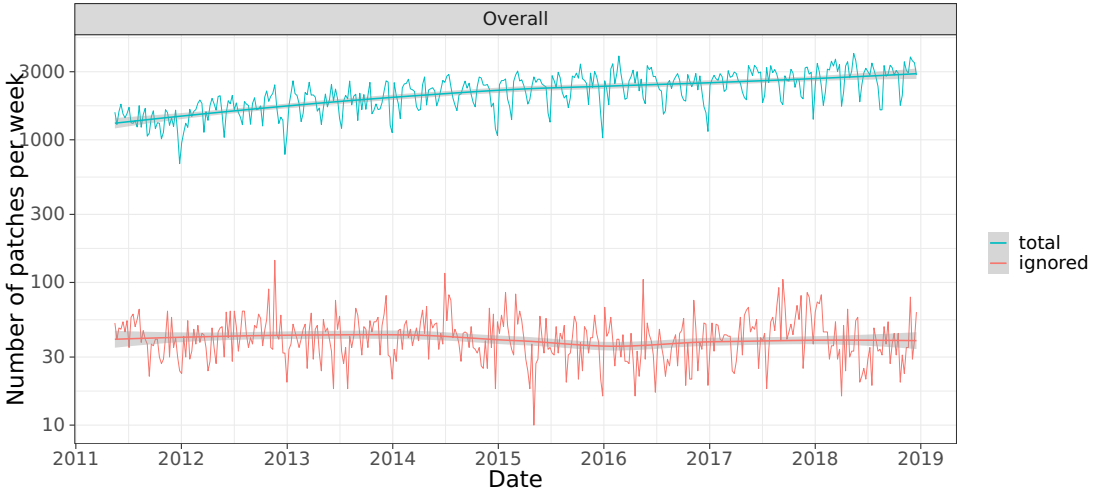
By the Numbers...

- lore.kernel.org lists 2011-2018: ø2.5% ignored patches
  - 2011: ø3.9%
  - 2015: ø2.1%
  - 2018: ø1.6%

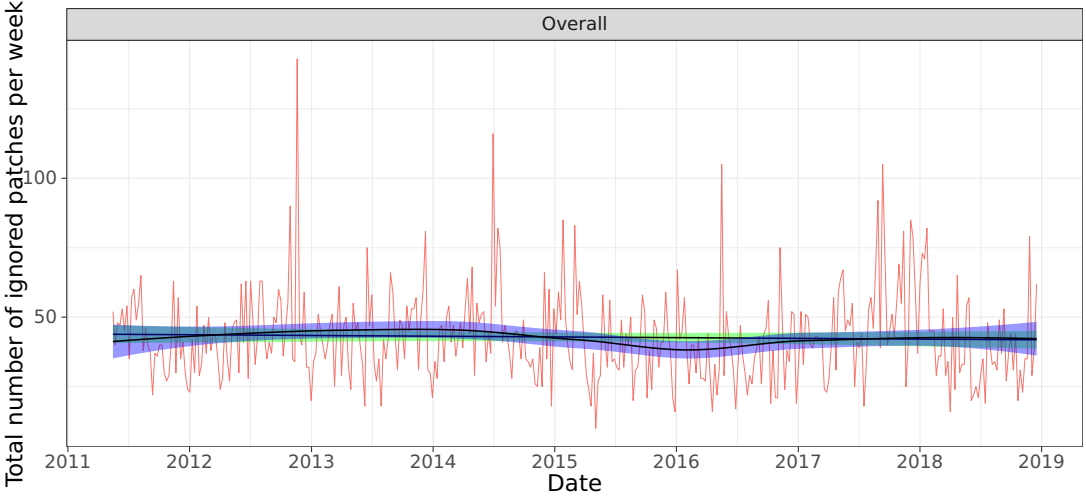
Evolution of ignored patches



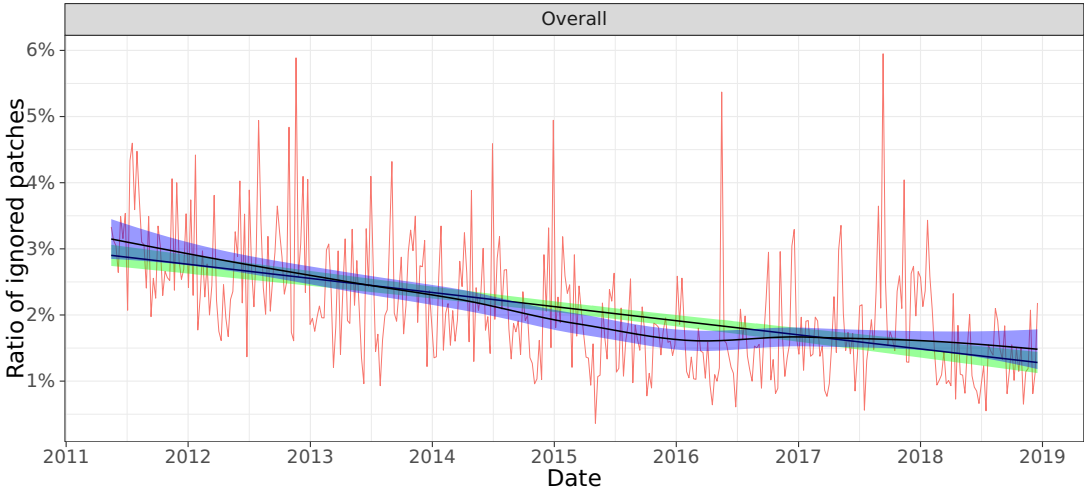
Evolution of ignored patches



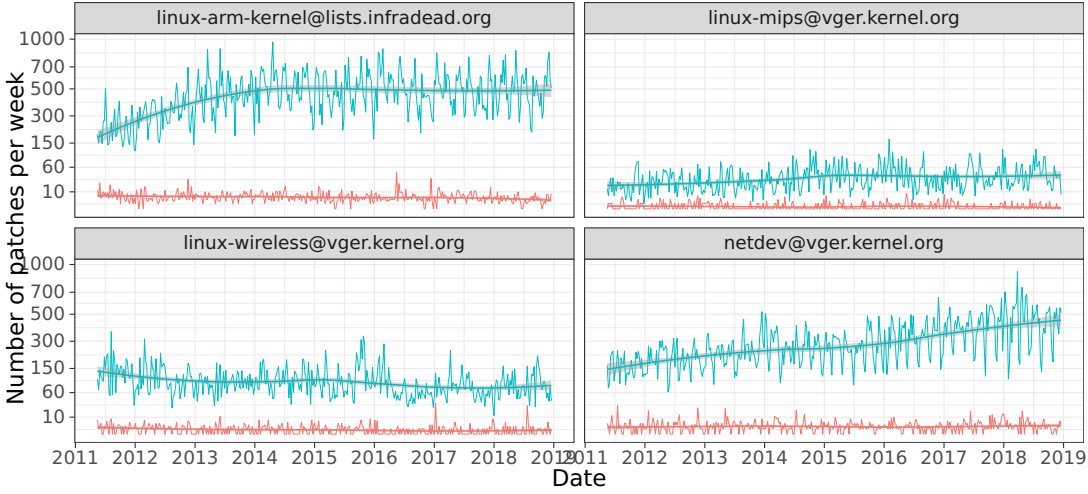
Evolution of ignored patches



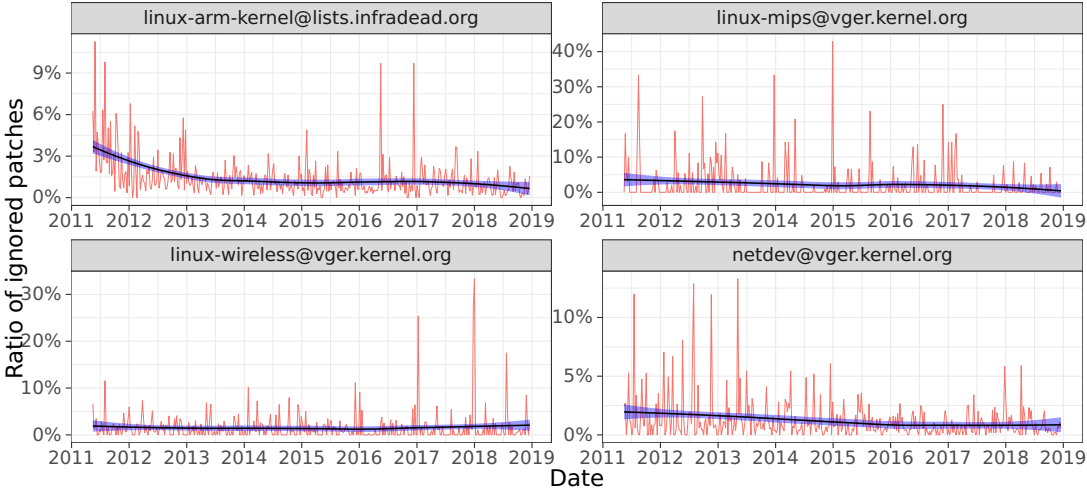
Evolution of ignored patches



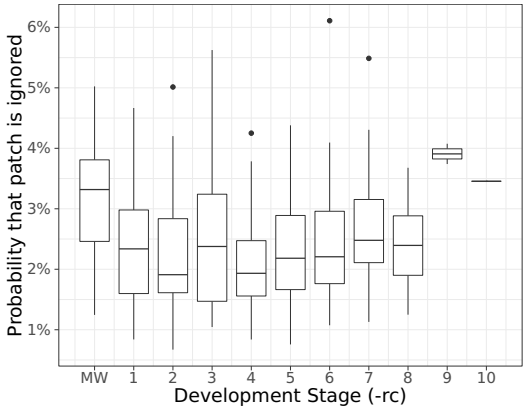
Evolution of ignored patches



Evolution of ignored patches



Does it matter *when* a patch is sent?

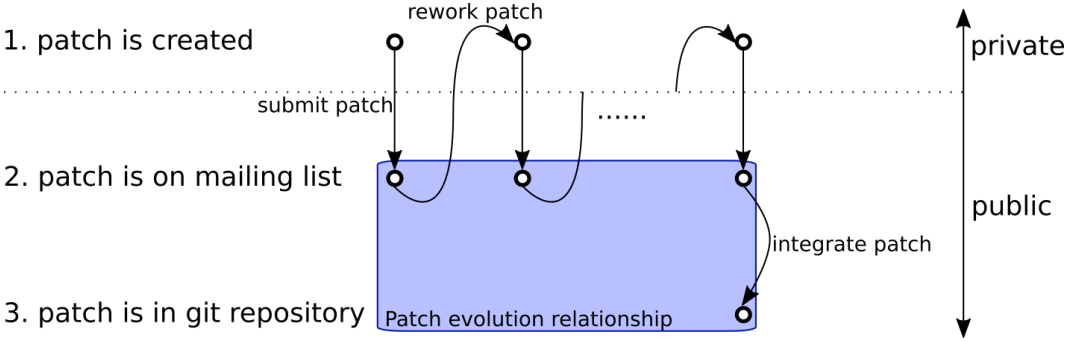


Distribution of ratio ignored/total patches grouped by linux kernel development stage

Insights

- Largely independent of the development stage
- Slightly higher chance of ignorance during merge window

Towards a formal model of the development process



Off-list Patches

Definition

An *off-list patch* is a patch that...

- ...has been included in Linus' git repository
- ...has never been sent to any public mailing list

Off-list Patches

Definition

An *off-list patch* is a patch that...

- ...has been included in Linus' git repository
- ...has never been sent to any public mailing list

Results

- Identified 80 commits with PaStA heuristics from v5.1-rc1..v5.1 (≈1800 commits)
- Manually assessed 60 commits and identified 24 off-list patch commits

Off-list Patches

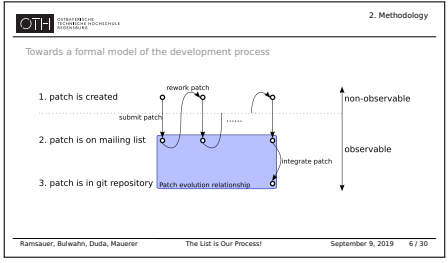
The obvious

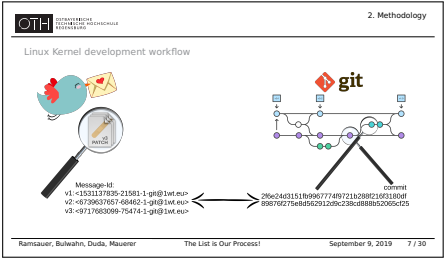
- Reverting patches is discussed on mailing list, the reverting patch is not sent.
- Very few patches from maintainers are actually off-list patches

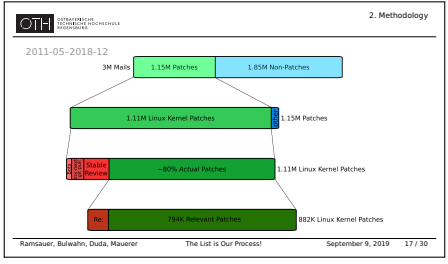
The less obvious

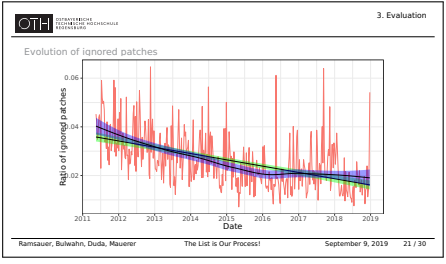
- Some off-list patches are clearly some security-related issues
- Patches from some subsystem maintainers are often off-list

```
commit c7084edc3f6d67750f50d4183134c4fb5712a5c8
Author: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
Date:   Fri Apr 5 15:39:26 2019 +0200

    tty: mark Siemens R3964 line discipline as BROKEN

    The n_r3964 line discipline driver was written in a different time, when
    SMP machines were rare, and users were trusted to do the right thing.
    Since then, the world has moved on but not this code, it has stayed
    rooted in the past with its lovely hand-crafted list structures and
    loads of "interesting" race conditions all over the place.

    After attempting to clean up most of the issues, I just gave up and am
    now marking the driver as BROKEN so that hopefully someone who has this
    hardware will show up out of the woodwork (I know you are out there!)
    and will help with debugging a raft of changes that I had laying around
    for the code, but was too afraid to commit as odds are they would break
    things.

    Many thanks to Jann and Linus for pointing out the initial problems in
    this codebase, as well as many reviews of my attempts to fix the issues.
    It was a case of whack-a-mole, and as you can see, the mole won.

    Reported-by: Jann Horn <jannh@google.com>
    Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>
    Signed-off-by: Linus Torvalds <torvalds@linux-foundation.org>
```


2. Methodology
Towards a formal model of the development process
1. patch is created
rework patch
submit patch
2. patch is on mailing list
3. patch is in git repository
integrate patch
Patch evolution relationship
non-observable
observable
Ramsauer, Bulwahn, Duda, Mauerer
The List is Our Process!
September 9, 2019
6 / 30


2. Methodology
Linux Kernel development workflow
git
Message-Id:
v1: <1521137835-21581-1-git@1wt.eu>
v2: <67396376 57-68462-1-git@1wt.eu>
v3: <9717683099-75474-1-git@1wt.eu>
commit
2f6e24d3151fb9967774f9721b2880216f3180df
898769275e8d56291209c238cd888b52005c125
Ramsauer, Bulwahn, Duda, Mauerer
The List is Our Process!
September 9, 2019
7 / 30


2. Methodology
2011-05-2018-12
3M Mails
1.15M Patches
1.85M Non-Patches
1.11M Linux Kernel Patches
1.15M Patches
Stable Review
~80% Actual Patches
1.11M Linux Kernel Patches
Re:
794K Relevant Patches
882K Linux Kernel Patches
Ramsauer, Bulwahn, Duda, Mauerer
The List is Our Process!
September 9, 2019
17 / 30


3. Evaluation
Evolution of ignored patches
Ratio of ignored patches
0.06
0.04
.02
2011
2012
2013
2014
2015
2016
2017
2018
2019
Date
Ramsauer, Bulwahn, Duda, Mauerer
The List is Our Process!
September 9, 2019
21 / 30

Thank you!

{ralf.ramsauer,wolfgang.mauerer}@othr.de
sebastian.duda@fau.de
lukas.bulwahn@gmail.com